Electronic Version
Stylesheet Version v1.1.1

Claims

[c1] 1. A method to verify a circuit design, comprising:
applying a bounded model checking technique to a first computer language representation of the circuit design and to a second computer language representation of the circuit design; and
determining a behavioral consistency between the first and second computer language representations.

[c2] 2. The method of claim 1, wherein applying a bounded model checking technique comprises:
jointly unwinding the first and second computer language representations of the circuit design to form a Boolean formula; and
checking a satisfiability of the Boolean formula.

[c3] 3. The method of claim 2, further comprising:
forming a first bit vector equation in response to unwinding the first computer language representation;
forming a second bit vector equation in response to unwinding the second computer language representation; and
comparing the bit vector equations.

[c4] 4. The method of claim 3, further comprising translating each bit vector equation into a SAT instance.

[c5] 5. The method of claim 2, wherein checking the satisfiability of the Boolean formula comprises applying a SAT procedure.

[c6] 6. The method of claim 5, wherein comparing the bit vector equations comprises determining a consistency of the bit vector equations.

[c7] 7. The method of claim 5, further comprising extracting a counterexample in response to the bit vector equations being inconsistent.

[c8] 8. The method of claim 2, further comprising continuing to unwind the computer language representations to find a counterexample in response to the Boolean formula not being satisfiable.

[c9] 9. The method of claim 8, further comprising terminating the method in response to a completeness threshold of a potential counterexample being exceeded.

[c10] 10. The method of claim 2, further comprising extracting a counterexample in response to the Boolean formula being satisfiable.

[c11] 11. The method of claim 1, further comprising repeating

the method to remove any remaining inconsistencies.

[c12] 12. The method of claim 1, wherein the first computer language representation comprises a hardware description language.

[c13] 13. The method of claim 1, wherein the second computer language representation comprises a C type language.

[c14] 14. A method to verify a circuit design, comprising:
unwinding a first computer language representation of the circuit design to form a first bit vector equation;
unwinding a second computer language representation of the circuit design to from a second bit vector equation; and
comparing the first and second bit vector equations for consistencies.

[c15] 15. The method of claim 14, further comprising translating each bit equation into a SAT instance.

[c16] 16. The method of claim 14, further comprising extracting a counterexample in response to the bit vector equations being inconsistent.

[c17] 17. The method of claim 14, further comprising continuing to unwind the computer language representations in

response to the bit vector equations being inconsistent.

[c18] 18. The method of claim 14, wherein comparing the bit vector equations comprises translating the unwound computer language representations into a Boolean formula.

[c19] 19. The method of claim 18, further comprising determining a satisfaction of the Boolean formula.

[c20] 20. The method of claim 19, wherein determining the satisfaction of the Boolean formula comprises applying a SAT procedure.

[c21] 21. The method of claim 19, further comprising extracting a counterexample in response to the Boolean formula being satisfiable.

[c22] 22. The method of claim 19, further comprising continuing to unwind the computer language representations to find a counterexample in response to the Boolean formula not being satisfiable.

[c23] 23. The method of claim 14, further comprising terminating the method in response to a completeness threshold of a potential counterexample being exceeded.

[c24] 24. The method of claim 14, further comprising repeating the method to remove inconsistencies.

[c25] 25. The method of claim 14, wherein the first computer language representation is a hardware design language.

[c26] 26. The method of claim 14, wherein the second computer language representation is a C type programming language.

[c27] 27. A method to verify a circuit design, comprising:
finding a loop;
replacing the loop by an unwinding assertion in response to an unwinding limit for the loop being exceeded; and
duplicating a loop body in response to an unwinding limit for the loop being less than a predetermined limit.

[c28] 28. The method of claim 27, further comprising guarding a loop body with a loop condition in response to duplicating the loop body.

[c29] 29. The method of claim 28, further comprising propagating constants in response to guarding the loop body with a loop condition.

[c30] 30. The method of claim 28, further comprising incrementing an unwind counter in response to duplicating the loop body.

[c31] 31. The method of claim 27, further comprising removing the loop in response to a loop condition being false.

[c32] 32. The method of claim 27, further comprising terminating a loop unwinding procedure in response to not finding a loop.

[c33] 33. A system to verify a circuit design, comprising:
a processor adapted to apply a bounded model checking technique to a first computer language representation of the circuit design and to a second computer language representation of the circuit design; and
a software program to determine a behavioral consistency between the first and second computer language representations.

[c34] 34. The system of claim 33, wherein the processor is adapted to jointly unwind the first and second computer language representations of the circuit design to form a Boolean formula and to check a satisfiability of the Boolean formula.

[c35] 35. The system of claim 34, further comprising a SAT solver to check the satisfiability of the Boolean formula.

[c36] 36. The system of claim 33, further comprising:
a first bit vector equation formed in response to unwinding the first computer language representation;
a second bit vector equation formed in response to unwinding the second computer language representation;

and

means for comparing the bit vector equations for consistency.

[c37] 37. The system of claim 33, wherein the first computer language representation of the circuit design comprises a hardware description language.

[c38] 38. The system of claim 33, wherein the second computer language representation of the circuit design comprises a C type language.

[c39] 39. A computer-readable medium having computer-executable instructions for performing a method, comprising:

applying a bounded model checking technique to a first computer language representation of the circuit design and to a second computer language representation of the circuit design; and

determining a behavioral consistency between the first and second computer language representations.

[c40] 40. The computer-readable medium having computer-executable instructions for performing the method of claim 39, further comprising:

jointly unwinding the first and second computer language representations of the circuit design to form a

Boolean formula; and
checking a satisfiability of the Boolean formula.

[c41] 41. The computer-readable medium having computer-executable instructions for performing the method of claim 40, wherein checking a satisfiability of the Boolean formula comprises applying a SAT procedure.

[c42] 42. The computer-readable medium having computer-executable instructions for performing the method of claim 40, further comprising:
forming a first bit vector equation in response to unwinding the first computer language representation;
forming a second bit vector equation in response to unwinding the second computer language representation; and
comparing the bit vector equations.

[c43] 43. The computer-readable medium having computer-executable instructions for performing the method of claim 42, wherein comparing the bit vector equations comprises determining a consistency of the bit vector equations.

[c44] 44. The computer-readable medium having computer-executable instructions for performing the method of claim 43, further comprising extracting a counterexam-

ple in response to the bit vector equations being consistent.

[c45] 45. The computer-readable medium having computer-executable instructions for performing the method of claim 40, further comprising continuing to unwind the computer language representations to find a counterexample in response to the Boolean formula not being satisfiable.

[c46] 46. The computer-readable medium having computer-executable instructions for performing the method of claim 40, further comprising extracting a counterexample in response to the Boolean formula being satisfiable.